Homework #1

[ECE30021/ITP30002] Operating Systems

Mission

- Solve problem 1, 2, and 3
  - Solve the problems on Ubuntu Linux (VirtualBox) using vim and gcc.

- Submission
  - Submit a .tgz file containing hw1_1.tgz, hw1_2.c and hw1_3.c on HISNet.

    압축명령어 압축파일명 압축 대상

    "tar cvfz hw1_*<student_id>*.tgz hw1_1.tgz hw1_2.c hw1_3.c"

    (Do not copy&past but type the above command) → 압축해보고 풀어도 보기 가끔씩 에러남
  - After compression, please check the .tgz file by decompressing in an empty directory.

    "tar xvfz hw1_*<student_id>*.tgz"

    압축해제 해제 할 파일명

- Due date: PM 11:00:00, Mar. 12th

Honor Code Guidelines

- "과제"
  - 과제는 교과과정의 내용을 소화하여 실질적인 활용 능력을 갖추기 위한 교육활동이다. 학생은 모든 과제를 정직하고 성실하게 수행함으로써 과제에 의도된 지식과 기술을 얻기 위해 최선을 다해야 한다.
  - 제출된 과제물은 성적 평가에 반영되므로 공식적으로 허용되지 않은 자료나 도움을 획득, 활용, 요구, 제공하는 것을 포함하여 평가의 공정성에 영향을 미치는 모든 형태의 부정행위는 단호히 거부해야 한다.
  - 수업 내용, 공지된 지식 및 정보, 또는 과제의 요구를 이해하기 위하여 동료의 도움을 받는 것은 부정행위에 포함되지 않는다. 그러나, 과제를 해결하기 위한 모든 과정은 반드시 스스로의 힘으로 수행해야 한다.
  - 담당교수가 명시적으로 허락한 경우를 제외하고 다른 사람이 작성하였거나 인터넷 등에서 획득한 과제물, 또는 프로그램 코드의 일부, 또는 전체를 이용하는 것은 부정행위에 해당한다.
  - 자신의 과제물을 타인에게 보여주거나 빌려주는 것은 공정한 평가를 방해하고, 해당 학생의 학업 성취를 저해하는 부정행위에 해당한다.
  - 팀 과제가 아닌 경우 두 명 이상이 함께 과제를 수행하여 이를 개별적으로 제출하는 것은 부정행위에 해당한다.
  - 스스로 많은 노력을 한 후에도 버그나 문제점을 파악하지 못하여 동료의 도움을 받는 경우도 단순한 문법적 오류에 그쳐야 한다. 과제가 요구하는 design, logic, algorithm의 작성에 있어서 담당교수, TA, tutor 이외에 다른 사람의 도움을 받는 것은 부정행위에 해당한다.
  - 서로 다른 학생이 제출한 제출물간 유사도가 통상적으로 발생할 수 있는 정도를 크게 넘어서는 경우, 또는 자신이 제출한 과제물에 대하여 구체적인 설명을 하지 못하는 경우에는 부정행위로 의심받거나 판정될 수 있다.

Problem 0: Install Ubuntu on VirtualBox

- Enable Virtualization Technology in CMOS
  - https://www.qnap.com/ko-kr/how-to/faq/article/intel-vt-x%EC%99%80-amd-svm%EC%9D%84-%ED%99%9C%EC%84%B1%ED%99%94%ED%95%98%EB%8A%94-%EB%B0%A9%EB%B2%95/
  - On many computers, VT is enabled in the default setting.

메모리 32G 잡기

- Install Ubuntu on VirtualBox following guide
  https://mainia.tistory.com/2379, https://ghostweb.tistory.com/979
  - Download and install VirtualBox
    - https://www.virtualbox.org/wiki/Downloads
      - Install VirtualBox Extension Pack to access host USB devices
    - Windows Hangul user name can cause installation problem
  - Create a virtual machine on VirtualBox
  - Set the number of processors (e.g., 1 -> 4)
    - https://technote.kr/180
    - If you cannot modify the number of processors, check whether VT is enabled.
  - Install Ubuntu Linux on your virtual machine
    - Download .iso file from https://ubuntu.com/download/desktop
    - Recommended version: v20.04

Problem 0: Install Ubuntu on VirtualBox

- After installation
  - Install frequently used utilities
    - □ Open a terminal window (press CTRL+ALT+T)
    - □ Run the following command

```
sudo apt install gcc make perl vim
```

→ gcc 설치 명령어

  - Install Guest Addons
    - □ https://www.itzgeek.com/post/how-to-install-virtualbox-guest-additions-on-ubuntu-20-04/
    - □ https://www.manualfactory.net/11071
  - Installing Hangul
    - □ https://pstudio411.tistory.com/entry/Ubuntu-2004-%ED%95%9C%EA%B8%80-%EC%9E%85%EB%A0%A5-%EB%B0%A9%EB%B2%95

우분투에서 파이어폭스 사용해 직접접속 or 가상머신 노트북PC 공유 하기

  - Set a shared folder for file transfer to the host computer
    - □ https://m.blog.naver.com/PostView.nhn?blogId=leejk9592&logNo=221011462890&proxyReferer=https:%2F%2Fwww.google.com%2F
    - □ Resolving permission issue: http://daplus.net/virtualbox-virtualbox-%EA%B3%B5%EC%9C%A0-%ED%8F%B4%EB%8D%94-%EA%B6%8C%ED%95%9C/

FAQ

- VirtualBox does not support M1 processor, yet.
  - Use UTM and Ubuntu ARM64 instead (https://webnautes.tistory.com/1580)
- If the keyboard does not work while installing Ubuntu, try with other USB keyboard.
- Windows Hangul user id can cause installation problem.
  - In this case, temporarily create an English user id and install VirtualBox.

Problem 1: Practice Linux and VIM

- Read the following tutorials carefully to learn basic commands of UNIX/Linux and vim editor:
  - Learn UNIX in 10 minutes
    - https://networking.ringofsaturn.com/Unix/learnUNIXin10minutes.php
  - Vim tutorial on youtube (22 min.)
    - https://www.youtube.com/watch?v=GWo_MxMIJJ4
  - Learn vi/vim in 50 lines and 15 minutes
    - https://www.perlmonks.org/?node_id=333737

- Note!

  현재 커서 앞 / 뒤 에 insert

  - vim commands i, a, A, o, O, p, P, dd and yy are frequently used.
  - Search the Internet for the dot (.) command of vim.
    - It's a very convenient command that repeats previous command.

.vimrc file

- Using vi, create a text file ‘.vimrc’ in your home directory with the following content.

```
set nu : line number 표시
set tabstop=4 : tab을 칠 때마다 4칸만 띄우기
set ai : auto indentation (자동 들여 쓰기)
set background=dark : 배경화면 검정색
syntax on : (코드의 명령어 종류에 따라 색상을 다르게 해줌.
```

Problem 1: Practice Linux and VIM

- Mission

1. Write, compile and run the add.c program on Ubuntu.
   - □ vim add.c # Then, type the add.c program.
   - □ gcc add.c
   - □ ./a.out 현재디렉토리
2. Compress add.c and a.out into hw1_1.tgz, then decompress it in another directory. 압축된 파일명
   - □ tar cvfz hw1_1.tgz add.c a.out
3. Create a directory 'Test'. $ mkdir Test
4. Enter the 'Test' directory. $ cd test
5. Uncompress the hw1_1.tgz into the 'Test' directory.
   - □ tar xvfz ../hw1_1.tgz
6. Copy the hw1_1.tgz file to the shared folder
   - □ sudo copy cp hw1_1.tgz <shared_folder>
   - □ Then, check whether hw1_1.tgz is in the shared directory of the host computer
7. Shutdown the virtual machine
   - □ sudo shutdown now

add.c

```
#include <stdio.h>

int main()
{
    int a = 0, b = 0, sum = 0;
    printf("Input two integers: ");
    scanf("%d %d", &a, &b);
    sum = a + b;
    printf("%d + %d = %d₩n", a, b, sum);

    return 0;
}
```

Problem 2

- Write a program that prints the following information
  - # of CPU cores
  - CPU model name
  - Total memory
  - Average workload for previous 1min, 5min, and 15min

- Example

```
$ ./hw1_2
# of processor cores = 4
CPU model = Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz
MemTotal =  16647172
loadavg1 = 0.520000, loadavg5 = 0.580000, loadavg15 =
0.590000
```

Problem 2

- Guideline

  1. Explore the following files (e.g., open with vim editor) to find the above information
     - Open /proc/cpuinfo with vim and find information about CPU
     - Open /proc/meminfo with vim and find information about memory

       가상파일 (kernel 속에 있는 변수들의 file 형태로 보여줌) ; disk 상 존재X
     - Open /proc/loadavg with vim and find average workload

     * The files under /proc are virtual files showing kernel variables.

  2. Write a C program that retrieves the above information from /proc/* files.

     → OS 제공 (low-level)
     - Use UNIX file system calls (open(), close(), read() functions) rather than C standard functions.

     standard func: OS 제공하는 func 이용해 구현 ex) fopen, fread...

     C언어

/proc/cpuinfo

```
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 94
model name      : Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz
stepping        : 3
microcode       : 0xffffffff
cpu MHz         : 4001.000
cache size      : 256 KB
physical id     : 0
```

Problem 2

- Read the following documents.
  - open(): http://man7.org/linux/man-pages/man2/open.2.html
  - close(): http://man7.org/linux/man-pages/man2/close.2.html
  - read(): http://man7.org/linux/man-pages/man2/read.2.html
  - write(): http://man7.org/linux/man-pages/man2/write.2.html
  - If necessary, read other internet documents to understand UNIX file system call.

지나치게 low-level 값 읽기 불편

- To read a string to the end of line, use ReadTextLine() on the next page.
  - int ReadTextLine(int file_descriptor, char str[], int max_len);
    - Arguments
      - file_descriptor: a file descriptor returned by open()
      - str, max_len: a text buffer (and buffer size) to provide the result
    - Return value:
      - 0: success
      - EOF: no more text to read

- Use sscanf() to retrieve strings or numbers from a text line.

UNIX File I/O

```
#include <stdio.h>
#include <string.h>

#include <unistd.h>  POSIX 운영체제 API에 대한 Access를 제공하는 header file
#include <fcntl.h>   Linux 시스템에서 열려진 파일의 속성을 가져오거나 설정 할때 사용

#define FILE_NAME "test.txt"
#define LENGTH 128

int main()
{
        int fd = 0;

        // writing file
        char out_buffer[LENGTH] = { 0 };
        strcpy(out_buffer, "Hello, World!");

        printf("Writing ₩"%s₩" into file ₩"%s₩".₩n", out_buffer,
FILE_NAME);
                                                쓰기 전용으로 파일을 연다
        fd = open(FILE_NAME, O_CREAT | O_WRONLY,
S_IRWXU);                         만약 path name 파일이 존재하지 않을 경우 생성
소유자에게 읽기, if(fd < 0){
쓰기, 실행 권한을 준다 printf("Failed to open %s to write.₩n", FILE_NAME);
                return -1;
        }
        write(fd, out_buffer, LENGTH);
        close(fd);

        // writing file
        printf("Reading file ₩"%s₩".₩n", FILE_NAME);
        char in_buffer[LENGTH] = { 0 };
        fd = open(FILE_NAME, O_RDONLY);
        if(fd < 0){              읽기 전용으로 파일을 연다
                printf("Failed to open %s to read.₩n", FILE_NAME);
                return -1;
        }
        read(fd, in_buffer, LENGTH);
        close(fd);

        printf("Read ₩"%s₩" from file ₩"%s₩".₩n", in_buffer,
FILE_NAME);

        return 0;
}
```

ReadTextLine()

Read and fully understand this code.

```
// global variables
char buffer[BUFFER_SIZE];
int buffer_size = 0;
int buffer_pos = 0;

int ReadTextLine(int fd, char str[], int max_len)
{
    int i = 0;
    int j = 0;
    int ret = 0;        return 값 커서가 위치한 location, 실패시 -1 return
함수의 seek pointer(커서)를 조정하는 함수; 파일의 특정 부분을 읽고 싶을 때사용
    // if current position is 0, reset buffer size and pos
    if(lseek(fd, 0, SEEK_CUR) == 0)
        buffer_pos = buffer_size = 0;  →파일의 현재위치를 기준으로 offset을 계산

    while(j < max_len - 1){
        if(buffer_pos == buffer_size){
새로운 줄 읽을때 buffer[0] = 0;
If 조건 실행    buffer_size = read(fd,  buffer,  BUFFER_SIZE);
            buffer_pos = 0;return:읽어온 데이터의 byte 크기.
        }   ↳각줄에 각 데이터 하나 (char)
            → 읽고있는 데이터의 index

        if(buffer_size == 0){      read 실패
            if(j == 0)
                ret = EOF;
            break;
        }
        while(j < max_len - 1
        && buffer_pos < buffer_size){
            str[j++] = buffer[buffer_pos++];
            if(str[j - 1] == '₩0' || str[j - 1] == 10){
                                  =Null
                j--;            // to remove LF
                max_len = j;    // to terminate outer loop
                break;          // break inner loop
            }
        }
    }

    str[j] = 0;

    return ret;
}
```

Problem 3

- Write function ParseCommand() that parses command into arguments
  - void ParseCommand(char *command, int *argc, char *argv[]);
    - command: command string to parse.

      Ex) “ls –al”, “cat readme.txt”, etc.
    - argc: pointer to provide the number of arguments (output parameter)
    - argv: argument list (output parameter)
      - argv[i] should contain the address of command[$from$], where $from$ is the start index of the i^{th} argument.
        - Put ‘₩0’ to command[to], where to is the end index of the i^{th} argument.
      - Fill argv[argc] with NULL to indicate the end of argument list.

Problem 3

Example) Combined with main() on the next page,

```
// the underlined commands are typed by the user
$ ls -al
argc = 2
argv[0] = ls
argv[1] = -al
argv[2] = (nil)
$ tar cvfz homework.tgz *.c
argc = 4
argv[0] = tar
argv[1] = cvfz
argv[2] = homework.tgz
argv[3] = *.c
argv[4] = (nil)
$ quit
Bye!
```

Problem 3

```
#define MAX_CMD 2048
#define MAX_ARG 256
void ParseCommand(char *command, int *argc, char *argv[]);
int main()
{
   char command[MAX_CMD];
   command[0] = command[MAX_CMD-1] = 0;  // for safety

   int argc = 0;
   char *argv[MAX_ARG] = { NULL };

   while(1){
      printf("$ ");
      fgets(command, MAX_CMD - 1, stdin);
      command[strlen(command)-1] = 0;   // trim ₩r

      if(strcmp(command, "quit") == 0 || strcmp(command, "exit") == 0)
         break;

      ParseCommand(command, &argc, argv);                          // TO DO: implement this function

      printf("argc = %d₩n", argc);
      for(int i = 0; i < argc; i++)
        printf("argv[%d] = %s₩n", i, argv[i]);
      printf("argv[%d] = %p₩n", argc, argv[argc]);                 // argv[argc] must be NULL
   }

   printf("Bye!₩n");

   return 0;
}
```

Null값 포함 읽어들일 최대 문자수

읽어드린 문자열저장

문자열을 읽어들일 stream의 File 객체를 가리키는 Pointer

(+) stdin : standard in

→ 명령어 한줄 단위